Haptic Control of a Simplified Human Model with Multibody Dynamics

James J. Troy
The Boeing Company
jim.troy@boeing.com

ABSTRACT

This paper describes a haptically enabled human model controlled by multibody dynamics. The application implements a reduced degree of freedom dynamics model, which is a prototype for a system with a higher number of degrees of freedom. This work is meant to provide insight into the response of a constrained multibody dynamics system to real-time haptic interaction. The model is manipulated through virtual coupling attachment points, allowing the user to feel inertia and environment contact reactions through a force feedback device. A brief comparison is made between 3-DOF and 6-DOF haptic devices regarding depth of interaction achieved for this application.

1. INTRODUCTION

Traditional human modeling user interface tools can be difficult to use for complex posture adjustments, especially in confined spaces where contact with multiple objects is required. A more efficient interface that includes arm contact force feedback should be able to take advantage of insights the user already has about getting an arm into and out of confined configurations. Achieving this type of interaction requires developing a physically based model of the system and integrating it with a haptic (force feedback) interface device.

Purpose of this paper is to introduce a prototype articulated human model that demonstrates the type of interaction possible with multibody dynamics, interactively controlled by a haptic device. The implementation described here is an initial design with three degree of freedom (DOF) limbs constrained to move in a plane.

Motivation: The ultimate goal of this line of research is build a fully functional virtual environment for simulation. A big part of that is a human model that can interact realistically with objects in a virtual world. In an engineering environment, the primary use for a haptically enabled human modeling system is in the area of accessibility analysis for manufacturing and maintenance. This type of simulation environment would offer designers and analysts a more efficient way to answer basic ergonomics questions like "can a real mechanic get his/her arm in there when removing that part?"

The ability to interact with the environment without a complete graphical representation of the scene is an important aspect of a fully functional simulation system. Although this may seem like a minor consideration, real-life maintenance and assembly applications often have situations where the mechanic cannot view all aspects of a task. In a haptic simulation environment, the level of interaction should be deep enough that users can work effectively even with visual obstructions.

2. BACKGROUND AND PREVIOUS WORK

With a few exceptions, most human modeling software uses kinematically defined posture control. Inverse kinematics and complex interpolation schemes have been developed to give the user a very detailed level of control over every aspect of the figure's motion. Most of this type of work had its start in robotics research [2], and has been adapted to work with computer graphics animation. In order to make something look like it is responding to external influences, animators need to make a lot of kinematic adjustments and use special spline functions to make the motion look realistic. Using kinematics alone as the driver for a haptic simulation is very difficult.

Although inverse kinematics solutions for the arm can keep up with the motion of haptic position inputs without causing too much of a strain on performance, there is no direct way to produce reaction forces from a kinematic solution alone. Collision response, inertia, momentum, compliance, and gravity are some of the aspects of a realistic simulation that are not part of a kinematics-based solution[1]. Heuristics can be defined to simulate some types of forces, but these are special case solutions that tend to be incomplete and cumbersome to maintain.

Dynamics offers a better general purpose solution. In a physically based system, Newtonian mechanics are used as the basis for motion. This is a better alternative for haptics, since forces are already part of the calculation. Efficient formulation of the dynamic equations of motion allow simulation to take place at interactive rates.

Recently, interactive control of physically based human model simulation with non-force feedback input has been accomplished [7][4]. But this level of physically based, interactive control for articulated figures has not been fully explored with haptics as the interface mechanism.

In addition to an efficient dynamics model, another key aspect of physically based haptics simulation is an efficient collision detection and force generation algorithm. Collision forces must be computed at high rates (1000Hz) in order to maintain simulation and haptic device stability. Achieving high rates in increasingly complex environments is an active research subject [3][5].

1. In the strict definition, kinematics is the study of motion without regard to the forces required to achieve it.

3. PHYSICALLY BASED MODELING

Developing a motion generation procedure based on the laws of physics will be the focus of this section. The basic process involves four steps: 1. derive efficient equations of motion for a system of interconnected bodies, 2. include control forces and torques, 3. include collision forces and torques, and 4. solve the equations of motion using numerical integration methods.

3.1 Multibody Dynamics

The first tasks in deriving the equations of motion will be to set the scope of problem and to define the structure of the multibody system.

Since most human modeling applications in engineering design analysis involve situations in which arm motion is the primary focus, derivation of the dynamic equations for the arm will be the main topic of this discussion. A full human arm has seven articulated degrees-of-freedom[2], but for this application a simplified 3-DOF articulated arm model will be implemented with motion constrained to the sagittal plane (the motion plane seen from a side view).

In order to reduce the computational effort required to obtain a solution, generalized coordinates will be used to define the equations of motion. Generalized coordinates describe the system with the minimum number of equations necessary; one independent variable will be solved for each degree of freedom. In general, the equations will be more complex to derive than those described in Cartesian coordinates, but there will be far fewer for the computer to solve at run time. Figure 1 shows the generalized coordinates and dimensions for the 3-DOF arm.

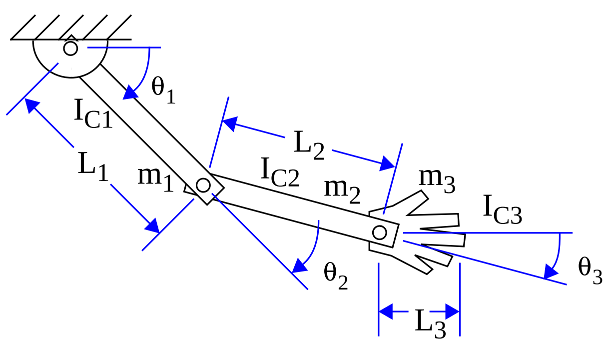

Figure 1. Planar 3-DOF arm coordinates and dimensions

The equations of motion will be second order ordinary differential equations (ODEs). For this application, the equations are derived by using the Lagrange method [8], which is a technique for describing the energy exchange between kinetic and potential forms. The Lagrange method begins by defining the motion of each body segment in terms of the partial differential equation described in Equation 1.

2. An unconstrained arm has seven primary degrees of freedom, not counting individual finger motions or shoulder translations.

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \tag{1}$$

Where q_i is the generalized coordinate, T is the scalar kinetic energy equation, V is the scalar potential energy equation, and Q is the virtual work. The resulting nonlinear equations are in the form, $A\ddot{X} = B$ and are then solved for in terms of the accelerations, $\ddot{X} = A^{-1}B$. Where $\ddot{X}$ is the vector of accelerations, A is a symmetric matrix of mass and inertia terms, and B is a function of the generalized velocities, positions, control forces, and collision reaction forces. Since the generalized coordinates are joint angles, all of the forces due to collisions and control inputs are converted into moments and appear in the B vector.

3.2 Control

For a haptics application, forces and torques transmitted to the end effector will be necessary.

Control of the arm is accomplished though a virtual spring/damper coupling attached to the wrist, as shown in Figure 2. This acts like a proportional-derivative (PD) controller [6], and will need to be tuned depending on the mass and inertial properties of the system. Other solution methods, like joint space control, can offer better goal position following, but requires an inverse kinematics calculation which adds to the computational overhead.

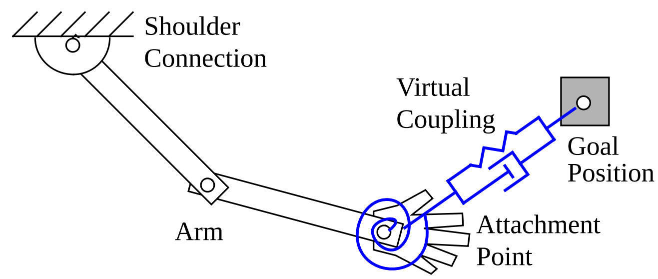

Figure 2. Arm with virtual coupling element

The equations for the control forces and torques transmitted to and from the haptics end effector are listed below.

$$U = -(X_{goal} - X_{attach})K_p - (\dot{X}_{goal} - \dot{X}_{attach})K_d \tag{2}$$

Where $U = [Fx, Fy, Fz, Tx, Ty, Tz]^T$, and K_p and K_d are the proportional and derivative gains, respectively. Note that Fx, Tx, and Ty are not part of the equations of motion for this 3-DOF arm. They can be removed from the equation above and used as a control mechanism for other aspects of the simulation. The forces and torques are described in terms of the wrist position (or ankle for a similarly configured 3-DOF leg), are then sent to the haptic device. For the PHANTOM®, the calculateForceFieldForce GHOST® functions are used to accomplish this.

3.3 Collision Detection

A simple polygonal-based system was initially developed to generate the appropriate reaction forces, but was eventually replaced with a more efficient force generation method that allows the object sizes and complexity to be scaled effectively.

The system currently uses a voxel based collision detection and contact force generation algorithm called Voxmap PointShell™ (VPS) [5]. This method defines each object as a collection of voxels and surface points. When the relative motion of two objects places a surface point of one object in the same volume as a voxel of the other, a contact event is detected and a penalty based force is generated.

Due to the voxelization process, the forces generated by VPS tend to be somewhat jaggy when just a few of the surface points are in contact, but this effect diminished when more of the points are in contact. The accuracy of collision position and force generation for this method is based on the size of the voxels and the number of surface points. If more memory and processing power are available then the size of the voxels can be reduced and the number of surface points increased. On the other hand, if more processing power is available, then a more complex dynamic model could be implemented instead.

One of the key challenges here is to balance the complexity of multibody dynamics computations with the accuracy of the voxel based collision detection method. For this application the system was tuned so that the amount of time spent in integrating the equations of motion is approximately the same amount as that spent generating collision forces.

Collecting Collision Results

In this application, collisions are processed in pairs instead of simultaneously. Objects with relative motion are fed two at a time into the collision detection algorithm and the forces from all collisions are collected and included in the equations of motion. The total number of pairs to be checked at each update is the combination:

$$C(n, 2) = n(n-1)/2 \tag{3}$$

For a system which includes a single three link arm and three external objects, the total number of pairs to be checked at each update is 15. As the system becomes more complex this can quickly get out of hand. For example, a system with 16 moving objects and links (which is reasonable for a fully articulated human model) the total would be 120 pairs!

Culling the list is important to achieving usable performance for larger systems. Taking advantage of joint limits to reduce the number of potential collisions between segments is a first step. Predictive algorithms, temporal coherence, and spatial partitioning can also help to reduce the number by helping to decided what needs to be checked and what does not. An application should also be prepared to experience the worst case scenario with all, or many, of the components coming into contact at the same time. Since the complexity of the current 3-DOF arm application is relatively low, a higher level part culling algorithm was not implemented. This means that the worst case collision situation is always in effect.

3.4 Solving the Equations of Motion

In order to solve the decoupled equations of motion, which are described in terms of the accelerations, $\ddot{X}$, the equations will need to converted into a series of *2N* first order equations in preparation for numerical integration. To solve the first order system, a constant time step numerical integration method is needed (a Runge-Kutta 4th order method was used here). The time step is adjusted so that the simulation can maintain stability in the case where a large numbers of collisions occur simultaneously. Adaptive time step methods are not used since the speeding up and slowing down of these methods causes inconsistent performance.

3.5 Other Implementation Issues

To increase application usability, a simple first order translation function can be used to pull the figure around the environment. It activates when an arm is fully extended and wrist forces reach a specified value. The only variables to adjust here are the activation force value and a velocity gain, which is a linear function of the wrist force.

Since the user has only one haptic device in this application for controlling multiple limbs, only one arm or leg can be independently controlled at any one moment in time. The active limb is controlled through attachment points on the wrists or ankles. When an arm is not actively controlled by the haptic device and multibody dynamics model, it stays in a locked posture at the previous position. An inverse kinematic function automatically positions inactive legs to keep the feet on the ground.

A limitation of this system is that the limbs are not dynamically coupled to each other through the torso. In a simulation environment with a single haptic device, simultaneous operation of both arms is not possible, so this limitation is not critical here. Dynamic coupling will become more important when two handed haptic applications are implemented or full body interaction is required.

A momentum transfer step is required when picking up objects. This is treated as an inelastic collision. Mass and inertia of the combined hand and object segment must be recalculated. The collision pairs list will also need to be updated to avoid calculating unnecessary collisions for the reconfigured system. It should be possible to pick up several objects using the same process for each new object, although this is not currently implemented.

As mentioned earlier, the complexity of the system model is limited by the processing power available. Since the haptic device must be updated at 1000Hz, it is best to try to match this rate with the model dynamics update rate. If the required update rate is not achieved, the forces sent to the haptics device are kept the previous values, and the computation falls into the next haptic refresh cycle. The result of unmatched updates is usually an unsatisfying washboard-like force effect.

4. RESULTS

A collection of functional test environments were assembled to evaluate the interaction capabilities of the human model with a static scene and movable objects. These consisted of part extraction and environment interaction tasks. One of the test cases, in which a hand tool is used to interact with the environment, is shown in Figure 3. The application was evaluated on two 6-DOF haptic devices: the PHANTOM 1.5/6DOF and 3.0/6DOF; as well as a 3-DOF PHANTOM Desktop. The application was hosted on a 250MHz dual processor SGI Octane.

Figure 3. Dynamics application with 6-DOF PHANTOM

Initial development was done with the 3-DOF PHANTOM Desktop device (which measures all six degrees of freedom, but only has force output on the three translational axes). Collisions with all parts of the arm and tool are transmitted to the users hand. Inertia and gravitational forces of the arm are also transmitted through the wrist of the model to the user's hand.

Attempting to extend the arm past the limit of reach produces a restoring force. If extended further, the force increases until it reaches a specified limit, after which the figure is translated along the direction of the force. This allows the user to drag the body to a desired location and seems to be a very natural way to interact with the model.

Although the PHANTOM Desktop haptic device does not have the ability to output torques at the end effector, the dynamics of the arm model allows some indirect artifacts of rotational motion to be perceived by the user. Translational motion is generated through the coupling of the wrist and arm segments, which provides important cues that wrist rotations are affecting the system. When simultaneously viewing the model motion on the screen and feeling the translational output it is possible to train yourself to accept this type of reaction as a partial substitute for true rotational force output.

When the same application was applied to a haptic device with 6-DOF force feedback the wrist torques were directly available for output. This gives the system a more natural quality feeling that comes closer to the goal of being able to explore the virtual environment without looking at the screen. As would be expected, the torque reaction is amplified by using extended hand tools like a hammer, pipe wrench, or tennis racket. The higher torque output of the larger of the two 6-DOF haptic devices (the PHANTOM 3.0/6DOF) gives a more convincing reaction in these amplified cases. After using the application on a device with 6-DOF force feedback, and feeling the interaction of the wrist torque due to collisions and rotational inertia of the various tools, going back to using the 3-DOF feedback device gives the user a sense that something is missing — or not working properly.

These comparisons are subjective in nature. In order to draw more objective conclusions with respect to performance benefits of 6-DOF vs. 3-DOF haptics, more formal studies will be needed.

5. CONCLUSIONS

An articulated human figure defined by a multibody dynamics model was presented in which limb motions are interactively controlled by a haptic device. The primary contribution of this type of system for human figure manipulation in a virtual environment is in its ability to allow more natural interaction modes. The results show that a human modeling application, enabled by interactive multibody dynamics and force feedback, can be made to respond with a heightened sense of realism and functionality, especially when using a 6-DOF haptic device.

REFERENCES

[1] Baraff, D., "Fast Contact Force Computation for Nonpenetrating Rigid Bodies." Proc. ACM SIGGRAPH 94 Conf., Orlando, FL, pp. 23-42, July, 1994.

[2] Craig, J.J, *Introduction to Robotics, Mechanics and Control*, Addison-Wesley, Reading, MA, 1986.

[3] Gregory, A., Lin, M.C., Gottschalk, S., Taylor, R., "A Framework for Fast and Accurate Collision Detection for Haptic Interaction." In the Proc. of IEEE Virtual Reality Conf., Houston, TX, Mar., 1999.

[4] Laszlo, J., van de Panne, M., Fiume, E., "Interactive Control for Physically Based Animation." Proc. ACM SIGGRAPH 2000 Conf., New Orleans, LA, pp. 201-208, July, 2000.

[5] McNeely, W.A., Puterbaugh, K.D., Troy, J.J., "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling." Proc. ACM SIGGRAPH 99 Conf., Los Angeles, CA, pp. 401-408, Aug., 1999.

[6] Ogata, K., *Modern Control Engineering*, 2nd ed, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[7] Troy, J.J., "Real-Time Dynamic Balancing and Walking Control of a 7-Link Planar Biped." Proc. ASME Design Engineering Technical Conf., Atlanta, GA, Sept., 1998.

[8] Wells, D.A., *Theory an Problems of Lagrangian Dynamics*, McGraw-Hill, New York, 1967.